UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/715,225	11/17/2003	Arun Kwangil lyengar	YOR920030488US1 (163-16)	5015

24336 7590 09/25/2008
KEUSEY, TUTUNJIAN & BITETTO, P.C.
20 CROSSWAYS PARK NORTH
SUITE 210
WOODBURY, NY 11797

EXAMINER	
TSAI, SHENG JEN	

ART UNIT	PAPER NUMBER
2186	

MAIL DATE	DELIVERY MODE
09/25/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/715,225
Filing Date: November 17, 2003
Appellant(s): IYENGAR ET AL.

Frank V. DeRosa
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 7/9/2008 appealing from the Office action mailed 1/17/2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Mater

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal Identifies the ground of rejections and the associated claims under rejection to be reviewed on appeal.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

US 2003/0172236	Iyengar et al.	09-2003
US 4,733,348	Hiraoka et al.	03-1988

US 2005/0128960	Chang et al.	06-2005

(9) Grounds of Rejection

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

>>> Claims 1-6, 10-12, 16-24 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Iyengar et al. (U.S. Patent Application Publication 2003/0172236, hereinafter referred to as Iyengar), and in view of Hiraoka et al. (US 4,733,348, hereinafter referred to as Hiraoka).

It is noted that, in the following claim analysis, those elements recited by the claims are presented using **bold** font.

As to claim 1, Iyengar discloses **in a system comprised of a plurality of storage elements** [figure 1 shows a system comprising a central cache (110), a remote server (104) and a plurality of processors (106-1~106-N) where each of the processor has a copy of cache as storage element, hence a plurality of storage elements], **a method for maintaining objects in the storage elements** [Methods and Systems for Distributed Caching in Presence of Updates and in Accordance with Holding Times (abstract); the corresponding objects in the storage elements are the contents of the caches] **comprising the steps of**:

maintaining information regarding which storage elements are storing particular objects [the central cache maintains local directories 110 which indicate the contents of local caches. A local directory maintains information about what objects may, but do not necessarily have to be, cached in the corresponding local cache. These local directories 110 allow a central cache to update local caches (paragraph 0030); the central cache 102 stores information from at least one remote server 104. The central cache communicates with a plurality of processors 106 which contain local caches 108. The central cache contains information about what is stored in local caches 108. When cached data changes, the central cache 102 is notified. The central cache is then responsible for updating local caches 108 (paragraph 0026)] **in a consistency coordinator** [the central cache (figure 1, 110) is the corresponding consistency coordinator; The central cache communicates with the one or more local caches and coordinates updates to the local caches] **which communicates with the storage elements** [the central cache 102 stores information from at least one remote server 104. The central cache communicates with a plurality of processors 106 which contain local caches 108. The central cache contains information about what is stored in local caches 108. When cached data changes, the central cache 102 is notified. The central cache is then responsible for updating local caches 108 (paragraph 0026)];

responding to a request to update an object [in step 202 (figure 2), a request for an object is issued (column 5, lines 5-15)] **by using maintained information to determine which of the storage elements may store a copy of the object** [The central cache contains information about what is stored in local caches 108. When cached data changes, the central cache 102 is notified. The central cache is then responsible for updating local caches 108 (paragraph 0026); In step 304, the central cache coordinates cache updates. That is, the central cache updates all objects it has cached which have changed. In addition, the central cache consults its local directories 110 to see which local caches may contain changed objects. Using local directories, the central cache 102 sends appropriate update messages to local caches (paragraphs 0039-0040)];

instructing the storage elements, which the consistency coordinator suspects store a copy of the object, to invalidate their copy of the object [The central cache communicates with the one or more local caches and coordinates updates to the local caches, including cache replacement (abstract); In step 304, the central cache coordinates cache updates. That is, the central cache updates all objects it has cached which have changed. In addition, the central cache consults its local directories 110 to see which local caches may contain changed objects. Using local directories, the central cache 102 sends appropriate update messages to local caches (paragraphs 0039-0040); When cached data changes, the central cache 102 is notified. The central cache is then responsible for updating local caches 108 (paragraph 0026); it is to be understood that the term "update," as used herein, is meant not only to include changing the value of a data object in a cache but also may include invalidating the data object or performing some other operation on the object. The central cache may communicate remotely with processes running either on the same processing node or on different processing nodes. That way, several applications running on different processing nodes may communicate with the same cache (paragraph 0010)]; **and delaying an updating of the object until it is determined that each storage element instructed to invalidate a copy of the object has either (i) acknowledged that it is not storing a valid copy of the object** [Iyengar: Updates to cached objects may go through the central cache. In order to update a cache object, the central cache may communicate with the local caches to make sure that all copies are invalidated or updated (paragraph 0011); thus the central cache delays the updating of an object until after it communicates communicate with the local caches to make sure that all copies are invalidated or updated; further, Hiraoka also teaches this aspect: A purge request source processor commonly supplies a purge request signal to other processors so as to cause them to perform TLB purge operations. A purge end signal sent back from other processors is stored in flip-flops in the source processor in units of processors. The source processor detects the end of TLB purge operations of all processors, in accordance with the statuses of the flip-flops (abstract)] **or (ii) been deemed un-responsible** [taught by Hiraoka, see below].

Regarding claim 1, it is noticed that the two limitations (i) and (ii) of claim 1 are recited based on an **either** (i) **or** (ii) condition, and it is typically recognized that it is suffice for a prior art to teach **either one** of the (i) and (ii) limitations to read on the claim limitations. Although it is understood that it suffices for a prior art to teach **either one** of the (i) **or** (ii) limitations to read on the claim limitations, as is the case of Iyengar's teaching of the (i) limitation, the Examiner determines to also teach the (ii) limitation, in the spirit of compact prosecution, as a preemptive response to Appellants' potential argument that the invention's intention is to have **both** (i) **and** (ii) limitations considered.

Regarding claim 1, Iyengar does not explicitly teach delaying an updating of the object until (ii) storage elements are deemed unresponsible.

However, it is well known in the art that a time-out mechanism is commonly used to avoid an infinite waiting period caused by a non-responsive element whose acknowledgement would never arrive. Without such a time-out mechanism, any non-responsive element would prevent the ensuing operations from being executed and cause the system to hang up.

Further, Hiraoka teaches in the invention "Virtual-Memory Multiprocessor System for Parallel Purge Operation" a scheme to send a purge command to a plurality of processors [as shown figure 3] to remove/invalidate a page from a Translation Lookaside Buffer (TLB) [abstract]. Specifically, with respect to claim 1, Hiraoka teaches **delaying an updating of the object until it is determined that each storage element instructed to invalidate a copy of the object has either (i) acknowledged that it is not storing a valid copy of the object** [A purge request source processor commonly supplies a purge request signal to other processors so as to cause them to perform TLB purge operations. A purge end signal sent back from other processors is stored in flip-flops in the source processor in units of processors. The source processor detects the end of TLB purge operations of all processors, in accordance with the statuses of the flip-flops (abstract); thus the acknowledgement from each storage element is stored and recorded in a corresponding flip-flop, and The source processor delays the updating until after it detects the end of TLB purge operations of all processors, in accordance with the statuses of the flip-flops] **or (ii) been deemed unresponsible** [The above operation can be performed when all the processors 20_0 through 20_3 are present. However, when the processor 20_3 is not present, the following operation is performed. The signal 48_3 representing that the processor 20_3 is not present is set at logic "1". The signal 48_3 of logic "1" is supplied to the OR gate 42_3. The OR gate 42_3 supplies the dummy TLB purge end signal to the AND gate 43. If the processor 20_3 is not present, the processor 20_0 can detect that all the TLB purge operations of the processors 20_0 through 20_2 are completed (column 4, lines 41-50). Note that processor 20_3 is the non-responsible element while processors 20_0 through 20_2 are responsive elements].

Therefore it would have been obvious for persons of ordinary skills in the art at the time of the applicant's invention to also take into consideration the situations where elements may be non-responsible, as demonstrated by Hiraoka, and to incorporate it into the existing scheme disclosed by Iyengar, in order to avoid the incidents that non-responsive elements would prevent the execution of the ensuing operations and cause a system to hang up.

As to claim 2, Iyengar teaches that **the step of maintaining information includes maintaining information regarding which storage elements are storing particular objects in the consistency coordinator** [The central cache contains information about what is stored in local caches 108. When cached data changes, the central cache 102 is notified. The central cache is then responsible for updating local caches 108 (paragraph 0026); In addition, the central cache consults its local directories 110 to see which local caches may contain changed objects. Using local directories, the central cache 102 sends appropriate update messages to local caches (paragraph 0040)].

As to claim 3, Iyengar teaches that **the consistency coordinator includes multiple nodes** [figure 1 shows the central cache (102), which is by itself one node, is connected to a remote server (104), which serves as another node to facilitate consistency coordination with remote storage elements; The central cache may communicate remotely with processes running either on the same processing node or on different processing nodes. That way, several applications running on different processing nodes may communicate with the same cache (paragraph 0010)] **and each node of the consistency coordinator stores information for a different set of objects** [since local caches require extra space and may thus in some situations be of limited size, it is preferred to have one or more methods for determining which objects to store in a local cache. Such methods, referred to as cache replacement policies, are described below in accordance with the present invention (paragraph 0029)].

As to claim 4, Iyengar teaches that **the storage elements include at least one cache** [figure 1 shows a plurality of nodes of processors (106-1~106-N) where each of the processor has a copy of cache as storage element].

As to claim 5, Iyengar teaches that **the storage elements are included in a distributed system** [figure 1 shows the configuration of a distributed system; in one aspect, a distributed caching technique of the invention comprises the use of a central cache and one or more local caches (paragraph 0010)].

As to claim 6, Iyengar teaches **the method as recited in claim 1, further comprising the step of obtaining a lock on the object to be updated before performing the update** [figure 5, steps 502, 504, 506 and 508; with respect to the locking or holding time issue, in another aspect, the invention provides techniques for adaptively determining such time values (paragraph 0013)].

As to claim 10, it recites substantially the same limitations as those recited in claim 1, and is rejected by the same reason as applied to claim 1. Refer to "As to claim 1" presented earlier in this section for details.

As to claim 11, it recites substantially the same limitations as those recited in claim 1, and is rejected by the same reason as applied to claim 1. Refer to "As to claim 1" presented earlier in this section for details.

As to claim 12, it recites substantially the same limitations as those recited in claim 3, and is rejected by the same reason as applied to claim 3. Refer to "As to claim 3" presented earlier in this section for details.

As to claim 16, it recites substantially the same limitations as those recited in claim 4, and is rejected by the same reason as applied to claim 4. Refer to "As to claim 4" presented earlier in this section for details.

As to claim 17, it recites substantially the same limitations as those recited in claim 1, and is rejected by the same reason as applied to claim 1. Refer to "As to claim 1" presented earlier in this section for details.

As to claim 18, it recites substantially the same limitations as those recited in claim 1, and is rejected by the same reason as applied to claim 1. Refer to "As to claim 1" presented earlier in this section for details.

As to claim 19, Iyengar et al. teach that **the system as recited in claim 18, further comprising a writer, which updates the object to be updated** [it is to be understood that the term "update," as used herein, is meant not only to include changing the value of a data object in a cache (paragraph 0010). It is noted that changing the value of a data object inherently requires a write operation, hence a writer].

As to claim 20, Iyengar et al. teach that **the writer resides on a same node as a storage element** [for example, the central cache which would update the cache data objects as shown in figure 1].

As to claim 21, it recites substantially the same limitations as those recited in claim 1, and is rejected by the same reason as applied to claim 1. Refer to "As to claim 1" presented earlier in this section for details.

As to claim 22, it recites substantially the same limitations as those recited in claim 1, and is rejected by the same reason as applied to claim 1. Refer to "As to claim 1" presented earlier in this section for details.

As to claim 23, Iyengar teaches that **the system as recited in claim 18, further comprising at least one content provider** [for example, the central cache or the remote server as shown in figure 1; for instance, a cache may be implemented as a server in a network (e.g., a cache server or proxy caching server in a World Wide Web or Internet environment) (paragraph 0009)].

As to claim 24, Iyengar teaches that **the content provider resides on a same node as a storage element** [for example, the central cache which would update the cache data objects as shown in figure 1].

As to claim 26, it recites substantially the same limitations as those recited in claim 4, and is rejected by the same reason as applied to claim 4. Refer to "As to claim 4" presented earlier in this section for details.

>>> Claims 7-9, 14-15 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Iyengar et al. (U.S. Patent Application Publication 2003/0172236, hereinafter referred to as Iyengar), in view of Hiraoka et al. (US 4,733,348, hereinafter referred to as Hiraoka), and further in view of Chang et al. (US Patent Application Publication 2005/0128960, hereinafter referred to as Chang).

As to claims 7-9, Iyengar in view of Hiraoka does not mention that **sending heart beat messages to obtain availability information to and from a storage element**.

However, Chang discloses in their invention "Method for Determination of Remote Adapter and/or Node Liveness" a heart beat message protocol for the determination of node liveness in a distributed data processing system [abstract; figures 6-8; paragraph 0017].

Chang teaches that using hear beat messages allows early detections of any failure component and prompt recovery operations to maintain high availability of system [Chang, paragraph 0003].

Therefore it would have been obvious for persons of ordinary skills in the art at the time of the applicant's invention to recognize the benefits using hear beat messages to identify faulty components as soon as possible, as demonstrated by Chang, and to incorporate it into the existing apparatus and method disclosed by Iyengar in view of Hiraoka, to further improve the availability and reliability of the system.

As to claims 14-15, they recite substantially the same limitations as those recited in claims 7-9, and are rejected by the same reason as applied to claims 7-9. Refer to "As to claims 7-9" presented earlier in this section for details.

As to claim 25, it recites substantially the same limitations as those recited in claims 7-9, and is rejected by the same reason as applied to claims 7-9. Refer to "As to claims 7-9" presented earlier in this section for details.

(10) Response to Arguments

Appellants' arguments have been fully and carefully considered with Examiner's answers set forth below.

(1) Appellant contends that **claim 1** is not obvious in view of Iyengar and Hiraoka because the references do not teach the limitation "delaying an updating of the object until it is determined that each storage element instructed to invalidate a copy of the object has **either** (i) acknowledged that it is not storing a valid copy of the object **or** (ii) been deemed unresponsive." The Examiner disagrees.

First, it is noticed that the two limitations (i) and (ii) of claim 1 are recited based on an **either** (i) **or** (ii) condition, and it is typically recognized that it is suffice for a prior art to teach **either one** of the (i) and (ii) limitations to read on the claim limitations.

Second, the Iyengar reference does teach the (i) limitation, that is, "delaying an updating of the object until it is determined that each storage element instructed to invalidate a copy of the object has (i) acknowledged that it is not storing a valid copy of the object:

Iyengar teaches [Updates to cached objects may go through the central cache. In order to update a cache object, the central cache may communicate with the local caches to make sure that all copies are invalidated or updated (paragraph 0011)].

Thus, Iyengar clearly teaches that the central cache delays the updating of an object until after it communicates communicate with the local caches to make sure that all copies are invalidated or updated.

Therefore, Iyengar clearly teaches at least the (i) limitation of the **either** (i) **or** (ii) limitations recited in claim 1.

Third, although it is understood that it suffices for a prior art to teach **either one** of the (i) **or** (ii) limitations to read on the claim limitations, as is the case of Iyengar's teaching of the (i) limitation, the Examiner determines to also teach the (ii) limitation, in the spirit of compact prosecution, as a preemptive response to Appellants' potential argument that the invention's intention is to have **both** (i) **and** (ii) limitations considered.

The Examiner acknowledges that the Iyengar reference does not teach the (ii) limitation, that is, "delaying an updating of the object until it is determined that each storage element instructed to invalidate a copy of the object has (ii) been deemed unresponsive;" and the Examiner relied on the Hiraoka reference, which teaches both (i) and (ii) limitation, as explained below.

Fourth, regarding the (ii) limitation, Hiraoka teaches a scheme to send a purge command to a plurality of processors [as shown figure 3] to remove/invalidate a page from a Translation Lookaside Buffer (TLB) [abstract]. Specifically, Hiraoka teaches **delaying an updating of the object until it is determined that each storage element instructed to invalidate a copy of the object has either (i) acknowledged that it is not storing a valid copy of the object** [A purge request source processor commonly supplies a purge request signal to other processors so as to cause them to perform TLB purge operations. A purge end signal sent back from other processors is stored in flip-flops in the source processor in units of processors. The source processor detects the end of TLB purge operations of all processors, in accordance with the statuses of the flip-flops (abstract); thus the acknowledgement from each storage element is stored and recorded in a corresponding flip-flop, and The source processor delays the updating until after it detects the end of TLB purge operations of all processors, in accordance with the statuses of the flip-flops] **or (ii) been deemed unresponsible** [The above operation can be performed when all the processors 20_0 through 20_3 are present. However, when the processor 20_3 is not present, the following operation is performed. The signal 48_3 representing that the processor 20_3 is not present is set at logic "1". The signal 48_3 of logic "1" is supplied to the OR gate 42_3. The OR gate 42_3 supplies the dummy TLB purge end signal to the AND gate 43. If the processor 20_3 is not present, the processor 20_0 can detect that all the TLB purge operations of the processors 20_0 through 20_2 are completed (column 4, lines 41-50). Note that processor 20_3 is the non-responsible element while processors 20_0 through 20_2 are responsive elements].

Thus, Hiraoka teaches both (i) and (ii) limitations.

Fifth, in summary,

-- the Iyengar reference alone teaches at least the (i) limitation,

-- the Hiraoka reference alone teaches both the (i) and (ii) limitations,

-- it suffices for a prior art to teach only one of **either** (i) **or** (ii) limitations to meet the claim requirement,

-- regardless whether Appellants' intention is to have **both** (i) **and** (ii) limitations, or **either** (i) **or** (ii) limitation considered, the combination of Iyengar and Hiraoka teaches both (i) and (ii) limitations.

Sixth, regarding Appellants' argument that it is improper to interpret Hiraoka's Table Lookaside Buffer (TLB) as an "object," it is noticed that the claim language is completely silent on what constitutes an "object." It is further noted that MPEP 2111 states that "**Claims Must Be Given Their Broadest Reasonable Interpretation**." Thus, a TLB, being a "buffer," certainly qualifies as an "object" within the context of the language of claim 1.

Therefore, the Examiner's position regarding the patentability of claim 1 remains the same as stated in the previous Office Action.

(2) Appellant contends that **claim 10** is not obvious in view of Iyengar and Hiraoka because the references do not teach the limitation "delaying an updating of the object until it is determined that each storage element instructed to invalidate a copy of the object has **either** (i) acknowledged that it is not storing a valid copy of the object **or** (ii) been deemed unresponsive." The Examiner disagrees.

Appellants contend that claim 10 is not obvious in view of Iyengar and Hiraoka for the same reason as applied to claim 1.

However, the Examiner has explained in details in Part (1) of this section why claim 1 is obvious in view of Iyengar and Hiraoka, and therefore claim 10 is also obvious in view of Iyengar and Hiraoka. Refer to Part (1) of this section for details.

(3) Appellant contends that **claim 11** is not obvious in view of Iyengar and Hiraoka because the references do not teach the limitation "delaying an updating of the object until it is determined that each storage element instructed to invalidate a copy of the object has **either** (i) acknowledged that it is not storing a valid copy of the object **or** (ii) been deemed unresponsive." The Examiner disagrees.

Appellants contend that claim 11 is not obvious in view of Iyengar and Hiraoka for the same reason as applied to claim 1.

However, the Examiner has explained in details in Part (1) of this section why claim 1 is obvious in view of Iyengar and Hiraoka, and therefore claim 11 is also obvious in view of Iyengar and Hiraoka. Refer to Part (1) of this section for details.

(4) Appellant contends that **claim 17** is not obvious in view of Iyengar and Hiraoka because the references do not teach the limitation "delaying an updating of the object until it is determined that each storage element instructed to invalidate a copy of the object has **either** (i) acknowledged that it is not storing a valid copy of the object **or** (ii) been deemed unresponsive." The Examiner disagrees.

Appellants contend that claim 17 is not obvious in view of Iyengar and Hiraoka for the same reason as applied to claim 1.

However, the Examiner has explained in details in Part (1) of this section why claim 1 is obvious in view of Iyengar and Hiraoka, and therefore claim 17 is also obvious in view of Iyengar and Hiraoka. Refer to Part (1) of this section for details.

(5) Appellant contends that **claim 18** is not obvious in view of Iyengar and Hiraoka because the references do not teach the limitation "delaying an updating of the object until it is determined that each storage element instructed to invalidate a copy of the object has **either** (i) acknowledged that it is not storing a valid copy of the object **or** (ii) been deemed unresponsive." The Examiner disagrees.

Appellants contend that claim 18 is not obvious in view of Iyengar and Hiraoka for the same reason as applied to claim 1.

However, the Examiner has explained in details in Part (1) of this section why claim 1 is obvious in view of Iyengar and Hiraoka, and therefore claim 18 is also obvious in view of Iyengar and Hiraoka. Refer to Part (1) of this section for details.

(6) Appellants contend that **claims 7-9, 14-15 and 25** should be allowable due to their dependency from claims 1, 10-11 and 17-18, respectively, and due to the allegation that claims 1, 10-11 and 17-18 are each allowable over Iyengar in view of Hiraoka. The Examiner disagrees.

However, the Examiner has explained in Part (1) through (5) of this section why claims 1, 10-11 and 17-18 are obvious in view of Iyengar and Hiraoka, and therefore claims 7-9, 14-15 and 25 are not allowable simply based on their dependency from claims 1, 10-11 and 17-18. Refer to Part (1) through (5) of this section for details.

(11) Related Proceedings Appendix

None

/Sheng-Jen Tsai/ ST TFSA Examiner, Art Unit 2186
/Matt Kim/ Supervisory Patent Examiner, Art Unit 2186
/Vincent F. Boccio/ VFB Primary Examiner, Art Unit 2169 Appeal Specialist Technology Center 2100

September 17, 2008